Micropayment

Micropayment

Table of Contents

Chapter 1. Print On Demand

Definition

Definition SAX

SAX (= “Simple API for XML”) war die erste allgemein verfügbare XML-API für Java. Bis heute stellt SAX quasi das Fundament für alle weiteren Entwicklungen in Bezug auf Java-XML-APIs dar. Besonderes Merkmal von SAX ist dabei sein “stream” - basierter Ansatz: Genau wie eine normale Textdatei in Java mittels eines Input - Streams eingelesen wird, so wird auch eine XML - Datei dem SAX - Parser, der die SAX - API konkret implementiert, zugeführt. Demzufolge wird das XML - Dokument “Stück für Stück” eingelesen. Werden bestimmte XML - Elemente erkannt, werden Ereignisse ausgelöst. Vom Entwickler zu implementierende Ereignis - Handler reagieren dann entsprechend auf diese Ereignisse. Auf diese Weise kann die gesamte Struktur und die Daten eines XML - Dokumentes ausgewertet werden.

Sequentieller Zugriff

Nachteilig bei diesem Ansatz ist dabei die Tatsache, dass eine Navigation innerhalb des Dokumentes nicht möglich ist. Die Datei wird ja einfach von Anfang bis Ende eingelesen. Ist eine Stelle im Dokument “überlesen” worden, so kann auf sie nicht mehr im selben Durchgang ohne weiteres zugegriffen werden, wenn sie noch einmal gebraucht werden sollte. Es sei denn, die vom Entwickler implementierten Ereignis - Handler würden gleichzeitig Struktur und Daten des Dokumentes im Speicher abbilden, und so während des Abarbeitens des Dokumentes quasi eine Arbeitsspeicher - Kopie anlegen, auf die dann auch noch nachträglich zugegriffen werden kann. Dies ist jedoch relativ aufwendig.

Vorteile SAX

Trotz dieses recht schwerwiegenden Nachteils ist SAX bis heute noch in Gebrauch. Dafür gibt es vor allem einen Grund: Geschwindigkeit. SAX - Parser sind im allgemeinen sehr

schnell. Deswegen ist SAX für die Problemstellungen zu präferieren, wo Navigation innerhalb des XML - Dokumentes weniger wichtig ist, als vielmehr eine hohe Geschwindigkeit.

Vorteile DOM

DOM (= “Document Object Model”) verfolgt einen gänzlich anderen Ansatz. Nach erfolgtem Einlesen eines XML - Dokumentes steht dieses als DOM - Baum im Speicher zur Verfügung. Innerhalb des DOM - Baumes kann dann beliebig navigiert und sowohl auf Daten als auch die Struktur Einfluss genommen werden.

Geschwindigkeit DOM vs. SAX

Nachteilig hierbei ist die geringere Geschwindigkeit: Da erst ein kompletter Lesedurchgang erfolgen und dabei ein DOM - Baum im Speicher erzeugt werden muss, kann DOM in Bezug auf die Geschwindigkeit nicht mit SAX konkurrieren. Auch ist DOM durch durch die Abbildung des Dokumentes im Speicher natürlich auch speicherintensiver.